Beyond Swarm Intelligence: Building Self-Managing Systems Based on Pollination

Holger Kasinger, Bernhard Bauer
Institute of Computer Science, University of Augsburg
D-86135 Augsburg, Germany
{kasinger,bauer}@informatik.uni-augsburg.de

Abstract: Nature exhibits a fruitful inspiration source for building self-managing systems. The human body's autonomous nervous system, its reflex and healing system, or its immune system inspired the building of self-managing systems in the same way as biological systems as ant, termite, or bee colonies. Thereby, self-managing systems based on such biological, self-organizing systems mostly rely on Swarm Intelligence (SI). However, as there exist management problems to that swarm-intelligent, self-managing solutions are hard to apply, novel biological paradigms have to be studied, which are also self-organizing but do not rely on SI in order to achieve self-management. This paper describes such a novel paradigm, pollination of flowers, and demonstrates exemplary how to build self-managing systems based on this paradigm.

1 Introduction

The need to build self-managing systems is widely known and accepted, whether due to the increasing complexity problem, the total cost of ownership, or to provide the way forward to enable pervasive and ubiquitous computation and communications [Hor01, GC03, KC03]. However, building such self-managing systems, i. e. systems being self-configuring, self-optimizing, self-healing, and self-protecting (self-* properties), is a very complex task, as desired self-* properties are difficult to design (see [KB05]). But in recent years it has widely been recognized that nature exhibits a fruitful inspiration source for building self-managing systems, as it provides examples of communication mechanisms, architectures, or systems that have perfectly demonstrated self-managing capabilities over millions of years.

One of the most familiar natural paradigms for self-managing systems obviously is the human body's autonomic nervous system (ANS), which provided the inspiration for the Autonomic Computing (AC) initiative [Hor01] launched by IBM in 2001. The ANS is that part of the nervous system that controls the vegetative functions of the body, such as circulation of the blood, body temperature, and breathing rate, thus disburdening the human brain of dealing with these and many other low-level, yet vital, functions. This natural model of autonomy was adopted by the hierarchical reference architecture for autonomic computing systems [IBM05], recommending a self-managing system design with components being allocated a so-called autonomic manager for the local self-management, managed by centralized autonomic managers for the global, system-wide self-management.

Fortunately, as centralized managers even in self-managing systems have their drawbacks, systems not necessarily involve autonomic managers and managed entities for their self-management [MJ06]. Biological systems, like ant, termite, or bee colonies, achieve self-management through entirely different, often fully distributed and emergent ways of processing information. There is often no subsystem responsible for global self-* properties, these properties follow from some simple local behavior of the components typically in a highly non-trivial way. This way of achieving self-management includes phenomenons like self-organization [Hey03] and emergence [Hol99] and provided an inspiration for the Organic Computing (OC) initiative [MSvdMW04] in building self-managing systems.

Most of the self-managing systems based on such biological paradigms so far rely on Swarm Intelligence (SI) [BDT99], i. e. these systems are typically made up of a population of simple agents interacting locally with one another by means of their environment. In particular, agents communicate to one another merely by modifying their local environment (stigmergic communication [Gra59]), e. g. ants communicate to one another by laying down pheromones along their trails. These local interactions finally yield to the global self-* properties that make a system self-managing. Such swarm-intelligent, self-managing systems in the first instance address management problems suited for interactions between agents of only one logical type, e. g. ants.

However, there exist management problems that require interactions between different logical types of agents, what makes swarm-intelligent systems hardly to apply. Fortunately, biology also provides a well-suited paradigm for this class of management problems: pollination [KB06]. From the biology's point of view, pollination is an important step in the reproduction of seed plants. Thereby pollen grains – the male gametes – are transfered from the anther of a flower to the carpel of a flower, i. e. the structure that contains the ovule – the female gamete. Pollination is not to be confused with fertilization, which it may precede. From our point of view, the pollination paradigm will help to build future self-managing systems due to the self-* properties implied inherently by the pollination process between plants and insects. Thereby, plants and insects are different logical types of agents and no interactions take place between the group of plants as well as between the group of insects.

The rest of this paper is organized as follows. Section 2 presents an exemplary problem domain, where self-management concepts are strongly required but can not be delivered by biological paradigms relying on SI. Section 3 presents the pollination paradigm with its useful self-* properties and a meta model for self-managing systems based on this paradigm, while section 4 instances this meta model with the entities of the problem domain and presents an example scenario to illustrate the application. Finally, section 5 concludes and presents the current work status as well as an outlook on future work.

2 Motivation

Today's international airports offer a couple of non-stop ground services for aircraft handling at a ramp (the place for embarking and disembarking), such as "supplying aircraft with electric power", "starting the aircraft engines", "baggage, cargo and airmail loading and unloading", "aircraft cleaning", "supplying with drink water", "passengers boarding and deplaning", "aircraft towing", "aircraft moving", "aircraft air conditioning and heating", "catering", "transportation of crew", "filling up with aviation fuels", and "supervising the performance of all activities, including their coordination and chronological order" for instance. The supervision and management of all these activities is subject to the ground control, a central facility at an airport. The management requires state-of-the-art technology in accordance with exactly determined operating instructions and time schedules of the handling process along with taking account of the actual aircraft type. But ground control also has to cope with any conceivable disturbances, e. g. absent ground vehicles, accidents on the apron, delayed or different typed aircrafts, unavailable passenger bridges, occupied ramps due to delays, or other activities not finished properly.

To make things worse, these centralized ground controls will become a bottleneck and single point of failure to airports in future. For instance, in 1994, Munich International Airport had 200.000 aircraft movements, while in 2005 this amount was already doubled by 400.000 aircraft movements. As this trend will continue worldwide, the result is an increased flight density at airports, which causes the latter to expand in the same manner. This in turn boosts the management efforts of centralized ground controls evermore. Thus, ground control is clearly in need of new management approaches to cope with these future challenges.

Self-management approaches based on SI are not suitable for this class of management problem, as real-time interactions are required between the aircrafts (every aircraft possesses permanently changing parameters for its handling) and the ground vehicles[1]. SI-based solutions in its initial intention would only enable interactions between the aircrafts or between the ground vehicles but not between these two groups.

3 The pollination paradigm

3.1 Biological pollination process

The flower pollination paradigm is well-suited for the class of management problems described in the last section. The paradigm is based on the cross-pollination process by insects, e. g. bees. Thus, two logically different typed components are involved in the process: plants – more precisely the flowers of a plant as pollen source and pollen sink – and pollination vectors (pollinators) – agents carrying pollen from the source's anther to the sink's stigma, the receptive part of the carpel.

In order to become pollinated during bloom, the flowers of a plant need to attract pollinators that pick up and deliver pollen (grains) respectively. Therefore, flowers provide certain attraction cues that may be visual (showy petals or sepals with obvious shape, size, and color) or olfactory (volatile chemicals that diffuse and are carried by air movements

[1]Assuming that future ground vehicles (robots) are driving automatically and carry the ground personnel to their usage sites

through the environment). Note, the pollination paradigm is based on olfactory cues only. Not until a successful fertilization succeeds the pollination, a flower ceases to attract pollinators, as there is no need of further pollen grains.

As pollinators usually are intelligent enough to avoid the energy waste of behaviors that do not result in some kind of reward, a flower needs to reward an attracted pollination vector. Hence, the vector will perceive the reward, e. g. nectar, as a result of its visit and come back again or visit similar flowers of the same species nearby in order to obtain additional rewards. While collecting the reward, the vector also unconsciously picks up and delivers pollen grains. Vectors collect rewards as long as they have had enough or they can not find anymore. This ensures an effective pollination.

3.2 Natural self-* properties

The pollination process between plants and insects exhibits all required aspects for a self-organizing system [WH04]: It exposes an *increase in order* evoked by the attracting and rewarding, it is *autonomous* as it has no external control, it is *adaptable and robust w.r.t. changes* as it has no single point of failure and it is *dynamical*. Over the past millions of years this self-organization led to the following useful self-* properties.

Self-configuration: The evolutionary link between a plant species and its pollinators is responsible for a seamless incorporation of new plants and pollination vectors into the entire pollination system. A plant is incorporated as soon as a linked vector scents its fragrance. Vice versa, as soon as a vector scents a linked fragrance, the vector is incorporated itself.

Self-optimization: Rewarding takes place by the "first come, first served" principle. Thus, vectors carrying pollen faster from flower to flower will collect more reward as other vectors. In addition, flowers providing higher reward will be visited more often as flowers with less reward. Both speeds up the pollination process by different (learned) strategies within the components.

Self-healing: The loss of pollination vectors yields (to a certain extent) no significant disturbance of the pollination process, as other pollinators will pick up and deliver pollen grains instead of. The reason is, that flowers produce pollen as long as they get fertilized (or their bloom is over before respectively).

Self-protection: Reward is only provided to vectors that pick up or deliver pollen during their visit. Flowers are structured in such a manner, that no intruders can receive any reward without picking up or delivering pollen as a trade-off. Furthermore, intruding vectors only being on a "journey through" have no effect to the pollination process.

Self-adaptation: A plant (species) not adapting its attracting and rewarding to the available pollination vectors over the long run will finally die out. Vice versa, a vector (species) not adapting its behavior to the specific characteristics of the available plants will become extinct either.

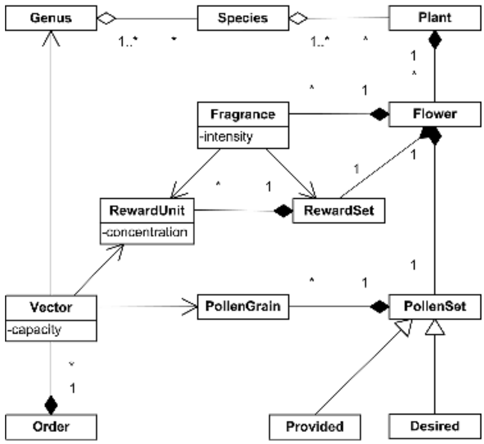

Figure 1: Meta model for self-managing systems based on pollination

3.3 Meta model for self-managing systems

Figure 1 depicts the meta model that has to be instanced by a self-managing system based on pollination. A formal specification of the meta model can be found in [KB06]. *Plants* are defined by its *genera* and *species*. Thereby, plants are allowed to be a member of one or more species, as well as a species is allowed to be a member of one or more genera at the same time. By contrast, a single *flower* of a plant is only allowed to be of a single genus and single species at the same time. The same holds for a pollination *vector* that belongs to exactly one *order* at the same time. A vector is a pollinator for only one or a few genera and can only pick up or deliver pollen grains from flowers of these genera.

A *pollen grain* within a *pollen set* is of the same species and genus as the flower it was produced by. Note, a pollen grain includes no more information, in particular no hint on the flower serving as addresser or addressee for it. A flower possesses a *provided* and a *desired* pollen set, each including a dynamically changing quantity of pollen grains. A *reward unit* within the *reward set* of a flower has a changing *concentration* and represents the reward (nectar drop) for a vector. The vector's *capacity* limits its ability to collect innumerable quantities of reward units and hence pollen grains.

A *fragrance* propagates the current reward conditions of a flower and therefore consists of all the information vectors need to decide to visit the flower: The genus, the species, the reward concentration, and the quantity of reward units (for pick up as well as delivery of pollen grains) provided by the flower. Additionally, an *intensity* is attributed to a fragrance, ensuring two natural aspects: Firstly, the temporal volatility of a fragrance, and secondly, the route guide for a vector. Note, according to nature, a fragrance consciously includes no information on the identity of the emitting flower. A vector follows a fragrance because it wants to receive an adequate reward, no matter from which flower of a certain species or genus. If the vector scents on its way to this flower another fragrance with better conditions, the vector may follow this new fragrance.

4 An autonomous aircraft handling system

4.1 Instancing the meta model

The application of the pollination paradigm of section 3 to the management problem described in section 2 first requires a mapping of real entities to classes of the meta model (an detailed instance model can be found in [KB06]).

Each *aircraft* is mapped on a plant, its *flight number* is mapped on a species, while each offered *ground service* by the airport – which is obtainable by aircrafts – is mapped on a genus. Each relevant aircraft facility (*baggage center*, ...) is mapped on a plant, too. To process the handling, flowers (*hatches* of an aircraft, *conveyor belts* of the baggage center, ...) emit *radio signals* as fragrances, that attract *ground vehicles* (mapped on vectors) of a certain *vehicle type* (mapped on orders). Just like in nature, such an attracted vector carries for example *pieces of baggage* (each mapped on pollen grains) from the provided pollen set of a flower (e.g. the *baggage set* of the baggage center) to the desired pollen set of another flower (e. g. the *freight by departure* of the aircraft's hatch) and hence are rewarded by the flowers with *money* (mapped on reward units) of the *supply of money* (mapped on the provided reward sets) of every flower.

4.2 Exemplary aircraft handling scenario

For an exemplary aircraft handling scenario (see Figure 2) we only focus on certain aspects of a single ground service, in order to illustrate the interaction principles of the paradigm.

Imagine an aircraft arriving at its dedicated ramp and requesting the ground service ”baggage loading and unloading”, for instance. Therefore, the hatch of the aircraft emits radio signals that include its genus (*baggage loading and unloading*), its species (*the aircraft's flight number*), the concentration of its reward set (*money per picked up / delivered piece of baggage*), and its provided reward set (distinguished between the *supply of money for picking up pieces of baggage* and the *supply of money for delivering pieces of baggage*). The intensity of the signal decreases with the distance of course. As soon as a baggage train within reach receives this signal, it starts evaluating, if the signal is of a compatible genus, if it has enough capacity left to collect some or all provided reward units, and if the provided reward unit concentration is high enough. On a positive result, the baggage train will follow the signal to its source, where the ground personnel can start to unload the baggage, while the baggage train collects the corresponding amount of reward units.

In the same manner, the conveyor belt of the baggage center emits signals, which include the same genus, the flight number of the aircraft, its own reward set concentration, and the distinguished, provided reward set. The concentration grade of the reward units is always up to the producing flower, i. e. if it is not high enough, less vectors may be attracted, which requires an increase of concentration. The baggage train that may already have unloaded the pieces of baggage of the aircraft may be attracted by this new signal and start following it in order to deliver its freight and to receive another reward. But also other baggage trains receiving this signal may be attracted and start picking up the provided pieces of baggage and deliver it to the aircraft – still emitting signals. As soon as the aircraft's hatch provides and desires no more pieces of baggage (this is the time of fertilization in nature as well as the time of ground service completion), it ceases to emit signals and no more baggage trains will be attracted. If all other ground services are completed in the same manner, the aircraft handling is finished and the aircraft is ready for departure.

Figure 2: Ground service "baggage loading and unloading"

5 Conclusion and outlook

This paper revealed that self-organizing and hence self-managing solutions relying on SI have interaction limits, which make them hard to apply to management problems requiring interactions between different logical agent types. Admittedly, approaches have been made to solve this class of management problems by combined SI solutions (see [VB05]), but these approaches mainly address optimization problems rather than global self-management. The paper in addition described the biologically-inspired flower pollination paradigm and showed its exemplary application to an aircraft handling scenario. Although the paradigm is –like swarm-intelligent systems – made up of a population of insects, it requires no interactions between these insects to achieve self-management. Self-management emerges merely by the self-organizational interactions between the two different logical types of agents, plants and insects, according to the biological example.

The pollination paradigm provides no blueprint for all management problems requiring interactions between different logical agent types. An application requires a mapping of plants, flowers, fragrances, pollen grains and vectors on appropriate entities of the problem domain. Beneath autonomous aircraft handling, one can think of autonomous manufacturing control, where robots (vectors) carry workpieces (pollen grains) to product machines (plants), or high bay warehouses with a similar behavior, for example.

To pave the way from the idea to real working, self-managing systems based on pollination, the next step will be the simulation and evaluation of the paradigm, in order to test and demonstrate the declared self-* properties. This will shed light on alterable system parameters that have to be adjusted in order to optimize the self-* properties. In a further step, a possible unintended emergent behavior of the system has to be bared, which is often a byproduct of biological paradigms.

References

[BDT99] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From natural to artificial sytems*. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, 1999.

[GC03] Alan G. Ganek and Thomas A. Corbi. The dawning of the autonomic computing era. *IBM Systems Journal*, 42(1):5–18, 2003.

[Gra59] Pierre-Paul Grasse. La reconstruction du nid et les coordinations interindividuelles chez bellicositermes natalensis et cubitermes sp. La theorie de la stigmergie: essai d' interpretation du comportement des termites constructeurs. *Insectes Sociaux*, 6:41–81, 1959.

[Hey03] Francis Heylighen. The science of self-organization and adaptivity. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers Co. Ltd., 2003.

[Hol99] John H. Holland. *Emergence: From Chaos to Order*. Perseus Books Group, 1999.

[Hor01] Paul Horn. Autonomic Computing: IBM's perspective on the state of information technology. Available at `http://www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf`, 2001.

[IBM05] IBM. An architectural blueprint for autonomic computing. Available at `http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf`, 2005.

[KB05] Holger Kasinger and Bernhard Bauer. Combining Multi-Agent-System Methodologies for Organic Computing Systems. In *3rd Int. Workshop on Self-Adaptive and Autonomic Computing Systems (SAACS 05) in Proc. Int. Workshop on Database and Expert Systems Applications (DEXA 2005), Copenhagen, Denmark*, pages 160–164, 2005.

[KB06] Holger Kasinger and Bernhard Bauer. Pollination – A Biologically Inspired Paradigm for Self-Managing Systems (accepted for publication). In *International Conference on Self-Organization and Autonomous Systems in Computing and Communications (SOAS 2006), Erfurt, Germany*, 2006.

[KC03] Jeffrey O. Kephart and David M. Chess. The Vision of Autonomic Computing. *Computer*, 36(1):41–52, 2003.

[MJ06] Robert Laddaga Márk Jelasity, Ozalp Babaoglu. Guest Editors' Introduction: Self-Management through Self-Organization. *IEEE Intelligent Systems*, 21(2):8–9, 2006.

[MSvdMW04] Christian Müller-Schloer, Christoph von der Malsburg, and Rolf P. Würtz. Organic Computing. *Informatik Spektrum*, 27(4):332–336, 2004.

[VB05] Paul Valckenaers and Hendrik Van Brussel. MAS Coordination and Control Based on Stigmergy. In *Proceedings of the 3rd European Workshop on Multi-Agent Systems (EUMAS 2005), Brussels, Belgium*, pages 507–508, 2005.

[WH04] Tom De Wolf and Tom Holvoet. Emergence Versus Self-Organisation: Different Concepts but Promising When Combined. In *Engineering Self-Organising Systems*, pages 1–15, 2004.